

# Этапы разработки программного обеспечения.

---

Прикладное программирование

# Цель урока

---

- Изучить:
- этапы разработки программного обеспечения;
- структуру спецификации программного продукта



# План урока

---

- Рассматриваемые вопросы:
- Структура программного обеспечения.
- Этапы разработки программного обеспечения.
- Постановка задачи.
- Рекомендуемая форма спецификации.

# Жизненный цикл программного обеспечения

---

- Первоначально термин **жизненный цикл (ЖЦ) продукции** был введен в середине 60-х годов 19 века голландскими специалистами по качеству **Дж. Ван Эттингером** и **Дж. Ситтингеном**.
- **ЖЦ продукции** включает три стадии:
- **Проектирование**
- **Изготовление**
- **Потребление**



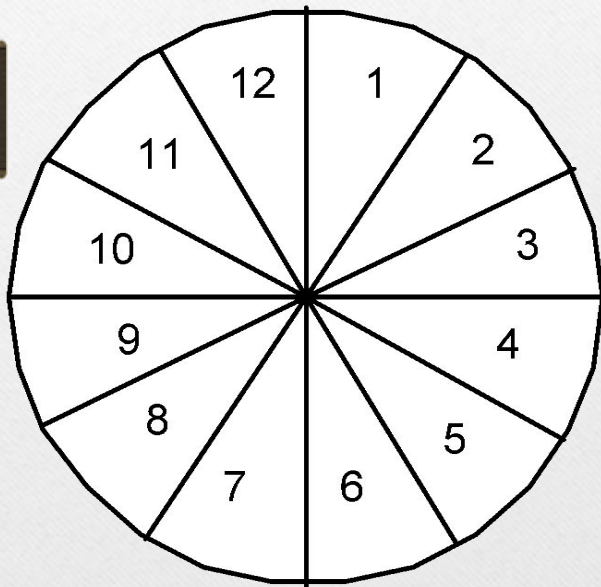
# Жизненный цикл программного обеспечения

---

- В развитии идеи Эттингера и Ситтинтена была разработана концептуальная модель взаимозависимых видов деятельности, влияющих на качество продукции, так называемая петля (спираль) качества (Quality Loop - QL).
- QL ориентирует на осознание того, что качество формируется на всех стадиях ЖЦ продукции. В дальнейшем QL была стандартизирована, в настоящее время международный стандарт ISO-9004-1-2000 поддерживает 12 стадий ЖЦ продукции

# Жизненный цикл программного обеспечения

---



1. Проектирование и разработка.
2. Подготовка производства.
3. Материально-техническое снабжение.
4. Производство.
5. Контроль и испытания.
6. Упаковка и хранение
7. Распределение и реализация.
8. Монтаж и ввод в эксплуатацию.
9. Техническое сопровождение.
10. Послепродажное обслуживание
11. Управление или восстановление после выработки ресурса
12. Маркетинг (поиск и изучение рынка)



# Программное обеспечение и программы

---

- Программой (программным модулем) называется последовательность команд, или операторов, которые необходимо ввести в оперативное запоминающее устройство компьютера, чтобы он мог обработать входные данные. Компьютер выполняет эту программу и выдаёт результат обработки входных данных (выходные данные).

# Программное обеспечение и программы

---

- Программное обеспечение состоит из нескольких совместно работающих программ (программных модулей), объединённых в программный комплекс, и документов, необходимых для разработки, сопровождения и эксплуатации программного комплекса.
- В простейшем случае программный комплекс может состоять из одной программы, при этом второй компонент программного обеспечения - документы - должен присутствовать всегда.



# Программное обеспечение и программы

## ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Программный комплекс

Программа 1

Программа 2

Программа N

Документы

Документы для разработки и сопровождения ПО

Документы по эксплуатации ПК

Другие документы

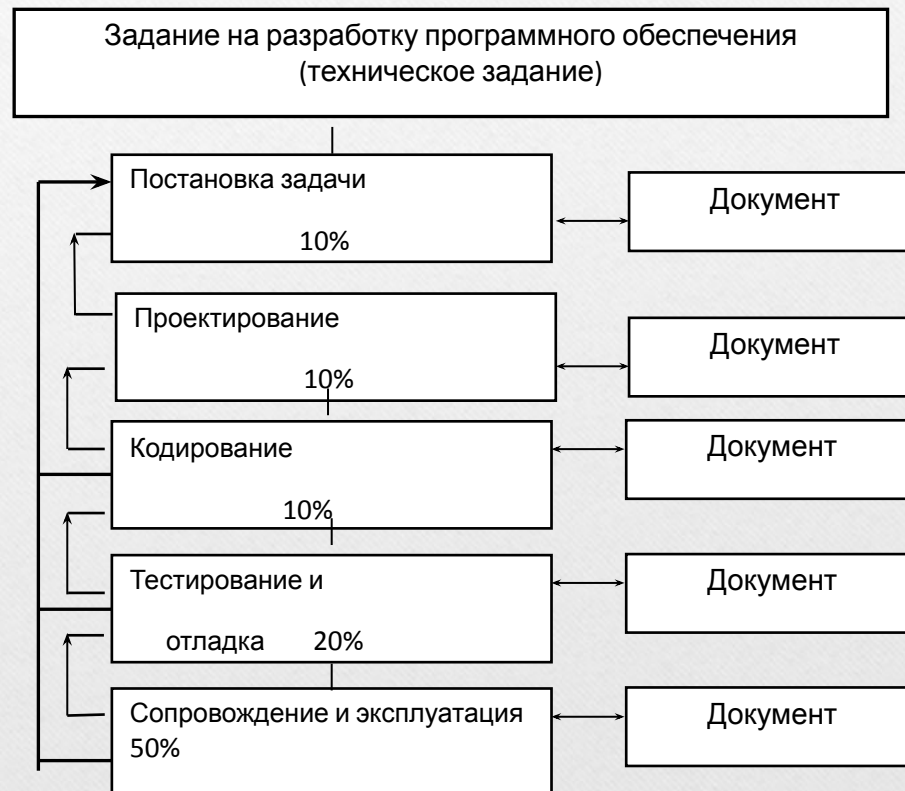
# Этапы разработки программного обеспечения

---

- Процесс разбивки программного обеспечения можно разбить на этапы.
- Работа над программным обеспечением начинается с выдачи задания на разработку программного обеспечения (технического задания).



# Этапы разработки программного обеспечения



# Этапы разработки программного обеспечения

---

- Постановка задачи.
- На этом этапе создаётся модель рассматриваемого явления, процесса и т.д. Характеристики и особенности этой модели отражаются в спецификации. Для подготовки спецификации используются определённые стандартные формы.



# Рекомендуемая форма спецификации

---

- Название задачи
- Приводится название решаемой задачи, программного комплекса, используемой системы программирования; указываются требования к компьютеру.

# Рекомендуемая форма спецификации

---

- Описание
- Подробно излагается условие задачи; описывается применяемая математическая модель для задач вычислительного характера, метод обработки входных данных для задач невычислительного (логического) характера и т.д.



# Рекомендуемая форма спецификации

---

- Управление режимами работы программы
- Формулируются основные требования к способу взаимодействия пользователя с программой (интерфейс пользователь - компьютер).

# Рекомендуемая форма спецификации

---

- **Входные данные**
- **Описываются входные данные, указываются пределы, в которых они могут меняться, значения, которые они не могут принимать, и т.д.**



# Рекомендуемая форма спецификации

---

- **Выходные данные**
- **Описываются выходные данные, сообщается о том, в каком виде они должны быть представлены - в числовом, графическом или текстовом, приводятся некоторые другие сведения.**

# Рекомендуемая форма спецификации

---

- Ошибки
- Перечисляются характерные ошибки при вводе данных, а, кроме того, возможные ошибки пользователя при работе с программой. Указываются способы диагностики ошибок и защиты от них на этапе проектирования, а также возможные реакции пользователя при совершении и ошибочных действий и реакция компьютера на эти действия.



# Рекомендуемая форма спецификации

---

- Примеры работы программного комплекса
- Приводится один или несколько примеров работы программного комплекса.

# Пример спецификации

---

- РАЗРАБОТАТЬ ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ РЕШЕНИЯ СЛЕДУЮЩЕЙ ЗАДАЧИ: ИМЕЮТСЯ ТРИ ЧИСЛА, ПРЕДСТАВЛЯЮЩИЕ СОБОЙ ДЛИНЫ ТРЁХ ОТРЕЗКОВ. ОПРЕДЕЛИТЬ, МОЖНО ЛИ ПОСТРОИТЬ ИЗ ЭТИХ ОТРЕЗКОВ ТРЕУГОЛЬНИК, И БУДЕТ ЛИ ОН ПРЯМОУГОЛЬНЫМ



# Пример спецификации

---

- Название задачи
- Треугольник.
- Название программы - `Triangl`.
- Система программирования - Турбо-Паскаль 7.0.
- Компьютер –AMD A6-3500 APU.

# Пример спецификации

---

- Описание
- Даны три вещественных положительных числа:  $a$ ,  $b$  и  $c$ .
- Определить:
- можно ли построить треугольник из отрезков, длина которых соответственно равна  $a$ ,  $b$  и  $c$ ?
- является ли этот треугольник прямоугольным?



# Пример спецификации

---

- Математическая формулировка задачи
- Условием существования треугольника является одновременное выполнение трёх неравенств:

$$b + c > a$$

$$c + a > b \quad (1)$$

$$a + b > c$$

# Пример спецификации

---

$$a * a = b * b + c * c$$

$$b * b = c * c + a * a \quad (2)$$

$$c * c = a * a + b * b$$

- Если все неравенства (1), выполняются и справедливо какое-либо из выражений (2), то треугольник будет прямоугольным.



# Пример спецификации

---

- Однако необходимо иметь в виду, что для переменных вещественного типа (real) использование логической операции отношения = (равно) даёт непредсказуемый результат из-за особенностей работы компьютера с числами данного типа. В связи с этим заменим равенства (2) на:

$$\begin{aligned} | a * a - (b * b + c * c) | / (a * a) < e \\ | b * b - (c * c + a * a) | / (b * b) < e \\ | c * c - (a * a + b * b) | / (c * c) < e \end{aligned} \quad (3)$$

# Пример спецификации

---

- Здесь  $\epsilon$  - малая величина. Гипотенузе прямоугольного треугольника будет, разумеется, соответствовать переменная, стоящая первой в левой части того из неравенств (3), которое выполняется (переменная  $a$ ,  $b$  или  $c$ ).



# Пример спецификации

---

- Управление режимами работы программы
- Для управления режимами работы программы использовать меню, в котором выбор режима осуществляется вводом номера (идентификатора). Работа с программой должна завершаться также с помощью меню.

# Пример спецификации

---

- **Входные данные**
- **Необходимо, чтобы  $a$ ,  $b$ ,  $c$  и  $e$  имели вещественный тип (real).**



# Пример спецификации

---

- Выходные данные
- На дисплей должна выдаваться справочная информация о назначении программы.
- После обработки входных данных, введённых пользователем, на дисплей должно выводиться одно из следующих сообщений:
  - 1) “Это не треугольник”
  - 2) “Это прямоугольный треугольник с гипотенузой, равной (указывается длина гипотенузы)”
  - 3) “Это не прямоугольный треугольник”

# Пример спецификации

Объект внешнего мира	Свойства объекта	Характеристики свойств	Связи между объектами или внутри объекта
Отрезки	Длины отрезков $a, b, c$	Число	Входные
Треугольник	Длины катетов, гипотенузы	Число	Выходные



# Пример спецификации

Объект задачи	Имя в программе	Характеристики			Характер использования: Вход/Выход/Константа
		Тип	Диапазон	Простая/Структура	
Длины отрезков	a, b, c	Веществ.	$[0..+\infty]$	Простая	Вход
Величина погрешности	e	Веществ.	$[0..1]$	Простая	Вход

# Пример спецификации

---

- Ошибки
- Примечание. При вводе численных значений типа `real` пользователь может ошибиться и ввести вместо числа комбинацию из числовых и нечисловых символов. Результатом этого является обычно зависание компьютера.



# Пример спецификации

---

- При вводе чисел  $a$ ,  $b$ ,  $c$  и  $e$  надо предусмотреть контроль:
  - 1) соответствия их типу `real`;
  - 2) знака вводимых чисел  $a$ ,  $b$  и  $c$ ;
  - 3) величины погрешности  $e$  ( $0 < e < 1$ ).
- При обнаружении ошибок программа должна выдавать соответствующие сообщения, которые могут сопровождаться звуковым сигналом, и предлагать повторить ввод.

# Пример спецификации

---

- Примеры работы программного комплекса

- Пример 1

$$e = 0.1$$

$$a = 5.01$$

$$b = 3$$

$$c = 4$$

- Это прямоугольный треугольник с гипотенузой, равной 5.01.



# Пример спецификации

---

- Пример 2

$$e = 0.1$$

$$a = 2$$

$$b = 3$$

$$c = 3$$

- Это не прямоугольный треугольник.

# Пример спецификации

---

- Пример 3

$$e = 0.1$$

$$a = 2$$

$$b = 3.5$$

$$c = 6$$

- Это не треугольник

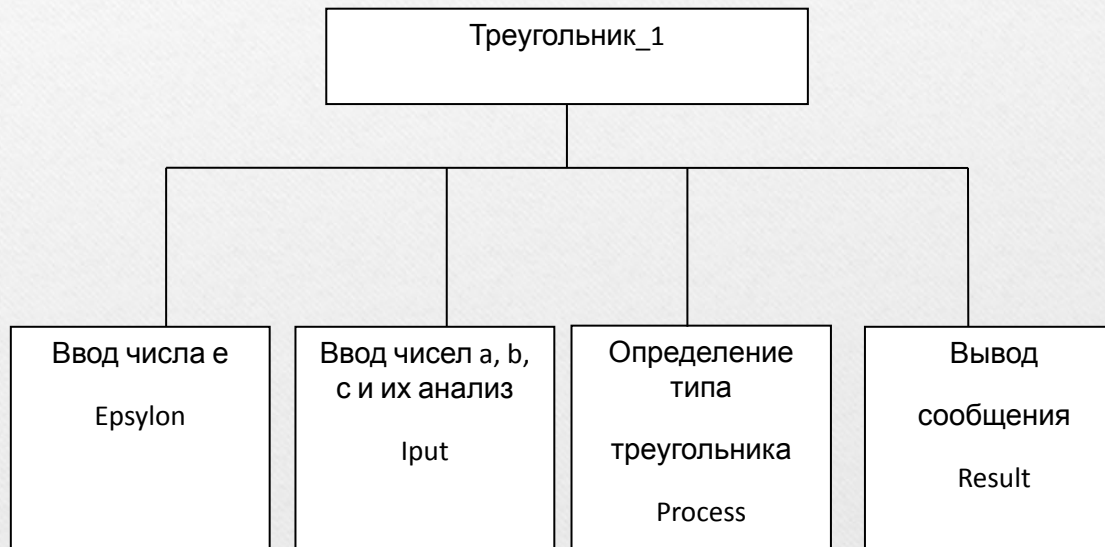


# Пример спецификации Процедуры и функции программы

Назначение процедуры (функции)	Имя в программе	Параметры		Примечания
		Входные: Имя, тип	Выходные: Имя, тип	
Ввод числа $\epsilon$	Epsilon			Процедура вызывается для ввода числа $\epsilon$ .
Ввод чисел $a$ , $b$ , $c$ и их анализ	Input			Процедура вызывается для ввода трех чисел $a, b, c$ и их анализа.
Определение типа треугольника	Process			Процедура вызывается для определения типа треугольника.
Вывод сообщения	Result			Процедура выводит результирующее сообщение.

# Пример спецификации

---





Д/з

---

- Изучить материал лекции, разработать спецификацию программного продукта, согласно варианта задания. См. Методические рекомендации по организации и выполнению самостоятельной работы С.р. №1.